



# SystemC CCI WG

## Directly Associating Parameters

P V S Phaneendra, CircuitSutra Technologies Pvt. Ltd  
August 2011



# Directly Associating Parameters

- **Objectives are to demonstrate the following:**
  - Parameter Searches (R3), and
  - Parameter Value Synchronization to address Different Names, Same Intended Meaning (UC9)

# Example Illustration

Get handles for owner modules' parameters using configuration broker *Name-based Look Up Access*

## Parameter\_Configurator

Create two *parameter\_owner* modules

Instantiate a *param\_value\_sync* class to synchronize the parameter values

Sends the list of selected cci\_param\_handles to the *param\_value\_sync* class

Param\_Name : clk\_freq\_Hz      Default Value : 1000 (Hz)

**Instance#1**  
**Parameter\_Owner**

Param\_Name : clock\_speed\_Hz      Default Value : 2000 (Hz)

**Instance#2**  
**Parameter\_Owner**

**Top\_Module**

# Example Illustration Cont'd

Configurator already stored the handles of the owner module `cci_parameters` to respective `cci_param_handle` through the broker

## Parameter\_Configurator

Select a list of cci-parameters using ***get\_param\_handle*** API using *Name-based Look Up Access* and pass it to the *param\_value\_sync*

A way to sync startup values of 2 parameters

With one cci-parameter as the reference, write it's default value to the other cci-parameter

## param\_value\_sync

**Param\_Name : clk\_freq\_Hz** Default Value : 1000 (Hz)

**Instance#1**  
**Parameter\_Owner**

**Param\_Name : clock\_speed\_Hz** Default Value : 2000 (Hz)

**Instance#2**  
**Parameter\_Owner**

## Top\_Module

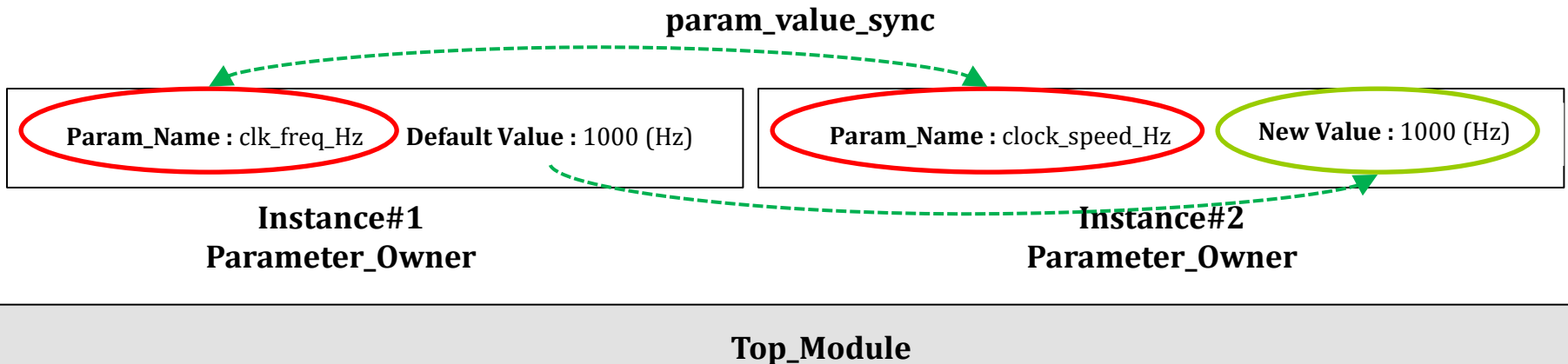
# Example Illustration Cont'd

Configurator already initialized handles of parameters to by sync'd

## Parameter\_Configurator

Synchronize values of the parameters (*before BEOE phase begins*)  
*syncValues (cci::cci\_param\_handle \_param\_handle\_1, cci::cci\_param\_handle \_param\_handle\_2)*

Register 'post\_write' callbacks on the selected parameters *clk\_freq\_Hz* and *clock\_speed\_Hz*



# Example Illustration Cont'd

Within BEOE phase, change value of *clk\_freq\_Hz* via CONFIGURATOR to 5000 (Hz)

Parameter\_Configurator

Registered Callback for *clk\_freq\_Hz* parameter is called and *clock\_speed\_Hz* value is synchronized with the *clk\_freq\_Hz* value

param\_value\_sync

Param\_Name : *clk\_freq\_Hz*

New Value : 5000 (Hz)

Instance#1  
Parameter\_Owner

Param\_Name : *clock\_speed\_Hz*

New Value : 5000 (Hz)

Instance#2  
Parameter\_Owner

Top\_Module

# Example Illustration Cont'd

Simulation starts

At 0ns, change value of *clk\_speed\_Hz* via CONFIGURATOR to 12000 (Hz)

Parameter\_Configurator

Registered Callback for *clock\_speed\_Hz* parameter is called and *clk\_freq\_Hz* value is synchronized with the *clock\_speed\_Hz* value

param\_value\_sync

Param\_Name : clk\_freq\_Hz

New Value : 12000 (Hz)

Instance#1  
Parameter\_Owner

Param\_Name : clock\_speed\_Hz

New Value : 12000 (Hz)

Instance#2  
Parameter\_Owner

Top\_Module

# Expected Output

## (ex11\_Directly\_Associateing\_Parameters.log)

SystemC Simulation

Info: top\_mod.param\_owner1: @0 s, [OWNER C\_TOR] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 1000

Info: top\_mod.param\_owner2: @0 s, [OWNER C\_TOR] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 2000

Info: top\_mod: @0 s, [TOP\_MODULE C\_TOR] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 1000

Info: top\_mod: @0 s, [TOP\_MODULE C\_TOR] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 2000

Info: param\_cfgr: @0 s, [CFGR C\_TOR] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 1000

Info: param\_cfgr: @0 s, [CFGR C\_TOR] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 1000

Info: sc\_main: Begin Simulation.

Info: param\_cfgr: @0 s, [CFGR within beoe] Within the BEOE phase

Info: param\_cfgr: @0 s, [CFGR within beoe] : Changing the 'clk\_freq\_Hz' of OWNER (1) to 5000 (Hz).

Info: top\_mod.param\_value\_sync: @0 s, [PARAM\_VALUE\_SYNC - post\_write callback] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 5000

Info: top\_mod.param\_value\_sync: @0 s, [PARAM\_VALUE\_SYNC - post\_write callback] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 5000



# Cont'd

Info: param\_cfgr: @0 s, [CFGR within beoe] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 5000

Info: param\_cfgr: @0 s, [CFGR within beoe] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 5000

Info: param\_cfgr: @0 s, @ 0 s

Info: param\_cfgr: @0 s, [CFGR] : Changing the 'clock\_speed\_Hz' of OWNER (2) to 12000 (Hz).

Info: top\_mod.param\_value\_sync: @0 s, [PARAM\_VALUE\_SYNC - post\_write callback] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 12000

Info: top\_mod.param\_value\_sync: @0 s, [PARAM\_VALUE\_SYNC - post\_write callback] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 12000

Info: param\_cfgr: @0 s, [CFGR] : Parameter Name : top\_mod.param\_owner1.clk\_freq\_Hz, Value : 12000

Info: param\_cfgr: @0 s, [CFGR] : Parameter Name : top\_mod.param\_owner2.clock\_speed\_Hz, Value : 12000

Info: sc\_main: End Simulation.